

MODULO 9

Medidas y Modelos de Sistemas

CONTENIDO:

–Conceptos sobre Mediciones en Sistemas.

–Generación de Modelos de Métrica.

–Métodos de mediciones de desempeños.

OBJETIVO DEL MÓDULO: Describir las distintas formas de diseñar medidas para evaluar el desempeño (Performance) del Sistema y detectar “cuellos de Botella” en el procesamiento de un Centro de Cómputos.

OBJETIVOS DEL APRENDIZAJE: Después de estudiar este módulo, el alumno deberá estar en condiciones de:

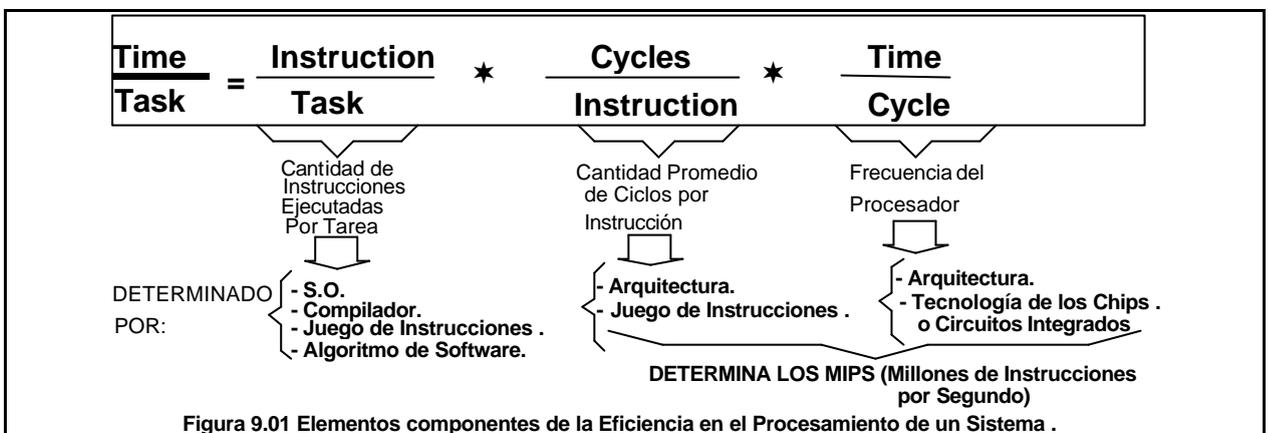
- Explicar y reconocer las distintas técnicas de medición utilizadas en Sistemas.
- Conocer la metodología de medición en sistemas.
- Conocer las funciones del Software que permite medir el desempeño del Sistema.
- Comprender y explicar como se organiza las mediciones y sus métodos.
- Conocer y explicar la terminología específica empleada en éste módulo.
- Entender los mecanismos de comparación utilizados en las mediciones.

9.0 Medidas y Modelos de Sistema (Métrica)

Hoy en día es necesario analizar el desempeño de un sistema de computación en el que juega un rol importante el aspecto en que el S.O. administra los recursos y su efectividad. En este módulo presentaremos un conjunto de herramientas que permiten modelar un problema y cuantificarlo para determinar el rendimiento del sistema.

Si bien en los primeros años de desarrollo de sistema computacionales, el hardware representaba un enorme costo, hoy no es así y el procesamiento de la información se transformó en un problema organizacional mas complejo en que se deben considerar, no solo el hardware y software, sino el ser humano y el ambiente con su respectivo Orgware.

Es normal que se busque eficiencia en el procesamiento. Todos los objetivos apuntan a que los trabajos sean realizados en el menor tiempo posible, o que los procesadores produzcan la mayor cantidad de tareas por unidad de tiempo o, que es lo mismo, para una dada tarea el menor tiempo de ejecución. Esto se considera como eficiencia cuando se utiliza el menor tiempo de ejecución por tarea.



Como se observa en la Figura 9.01, el primer término de la fórmula interviene el aspecto del Software en cuanto al Lenguaje de Alto Nivel utilizado para codificar el algoritmo que se pretende ejecutar en un dado ambiente computacional. Es de destacar que el Lenguaje de alto nivel se traduce en el Juego de Instrucciones del Procesador mediante un Compilador. Esta traducción de código juega un rol muy importante en la eficiencia de la ejecución del algoritmo. Si el traductor no optimiza el código objeto, seguramente que el tiempo de ejecución aumentará.

El S.O. también influye con su overhead, y por supuesto, la complejidad del algoritmo contribuye con lo suyo. De ésta forma se determina el número de instrucciones que deberá ejecutar el procesador en un dado tiempo.

El segundo y tercer término de la fórmula lo determina el Hardware. Los fabricantes tratan de optimizar para que se ejecuten la mayor cantidad de Instrucciones por segundo (MIPS¹) mejorando la tecnología de los Circuitos Integrados (Chips), aumentando la frecuencia de los relojes (MHz o GHz), Seleccionando la forma de ejecutar las instrucciones (CISC o RISC), buses más anchos etc. Esta es la razón por la que compiten ofreciendo mejoras en cada nuevo modelo.

Entonces la Eficiencia depende de la integración del Sistema: del Hardware y del software y en particular del algoritmo del Software de aplicación (si es de Tiempo Real, Operaciones Aritméticas, etc.) y de la modalidad operativa - organizacional.

La Eficiencia se mide. Para medir se suelen construir Benchmarks o algoritmos adecuados que permiten comparar o determinar el comportamiento de la Aplicación en ese ambiente computacional.

La pregunta obligada es ¿cómo mejorar la prestación?, Esto se logra con un Hardware más veloz, o sea utilizando las Arquitecturas multiprocesadores o RISC que buscan ejecutar en un solo ciclo de reloj la mayor cantidad de instrucciones posibles.

¹ MIPS significa Millones de Instrucciones Por Segundo.

También se observa que los Lenguajes de Alto Nivel integrados al S.O. son mas eficientes, como es el caso del C y el UNIX.

Pero aún queda un gran campo de evolución para el Software. Se investiga nuevos paradigmas, nuevas metodología, nuevos algoritmos, etc. Y los avances son muy lentos comparados con el hardware.

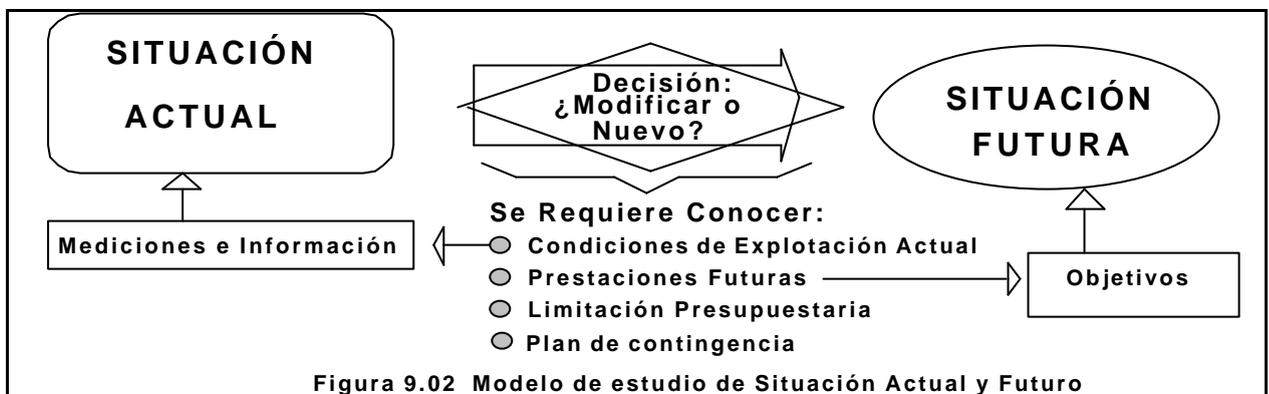
Objetivos:

- Determinar la efectividad del desempeño del Sistema (Hardware + Software + Ambiente).
- Analizar técnicas para lograr maximizar el rendimiento de los Sistemas.
- Estudiar el comportamiento del Sistema.

9.1. Estudios Cuantitativos

El estudio cuantitativo de sistemas permite comprender su funcionamiento y mejorarlos. Interesa una adecuada utilización del hardware y de los programas cuyo costo y complejidad van en aumento.

9.1.1. Áreas de aplicación de las Técnicas de Evaluación



Estudio de la situación actual:

En la Fig. 9.02 se presenta un modelo en que se deben considerar los ocho siguientes puntos.

1. Elección o modificación de las configuraciones o de los sistemas

- Se tiene una instalación que presenta "un Problema". Luego se decide que hacer: ¿Se modifica la instalación actual o se compra una nueva?
- Requiere que se conozcan las condiciones de explotación actual y las prestaciones que se desean obtener. A partir de las medidas que se hagan en los sistemas existentes se puede llegar a prever cual será el comportamiento de los nuevos sistemas.
- La previsión deberá ser lo más exacta posible ya que intervienen intereses económicos.

2. Contabilidad

- Se pueden repartir los costos entre un grupo de usuarios en función de los recursos que utilice realmente cada uno, si se conocen las tasas de utilización de los distintos recursos de una instalación.
- Las estadísticas que se obtengan podrán servir de guía para ampliar el sistema, modificar las condiciones de utilización y mejorar los servicios que se presten.

3. Optimización de programas

- Las prestaciones de los programas que ejecutan una tarea determinada pueden variar en proporciones considerables.
- Es fundamental es verificar la eficacia de los programas que se utilicen con frecuencia.
- La optimización de las prestaciones es importante en los programas que vayan a ser utilizados por un conjunto de usuarios. A través de medidas se pueden conocer las partes que más se utilizan.
- Los programas sufren modificaciones a lo largo de su existencia y es útil comprobar que dichas modificaciones no alteran sus prestaciones.

4. Diseño y construcción de los sistemas operativos

—Es útil tener datos cuantitativos sobre sistemas análogos al que se esté proyectando usar o se esté usando, tanto del funcionamiento interno como del entorno de trabajo, para establecer sus especificaciones y formular las hipótesis iniciales.

—En la fase de realización, las medidas permiten obtener los resultados esperados mediante evaluaciones y modificaciones sucesivas.

5. **Diseño del hardware**

—Los modelos y las medidas, más que para el diseño de los programas, son útiles para el hardware, puesto que cualquier modificación del cableado puede resultar muy costosa y larga de implantar.

—Es esencial conocer las características de la carga.

6. **Investigación sobre sistemas**

—El desarrollo de los modelos teóricos de programas y de sistemas y la comprensión del funcionamiento de los sistemas existentes, requiere un conocimiento profundo del comportamiento de estos programas y sistemas. Esto lleva a definir las dimensiones que caracterizan dicho comportamiento y a obtener sus valores numéricos por medio de medidas.

7. **Comportamientos:**

—Las Instalaciones son manejadas y operadas por seres humanos. Estos “imprimen” características al funcionamiento del Centro de Cómputos y muchas veces con solo observar “la situación o el contexto” se pueden aplicar programas de mejora a través de capacitación u organización del ambiente.

—Se debe observar al personal, el equipo de trabajo, las políticas, la Empresa, etc. Para evaluar el comportamiento.

8. **Simular:**

—Si la situación es muy compleja se puede simular los comportamientos o los parámetros mediante un software adecuado.

9.1.2. Métodos y medidas de evaluación

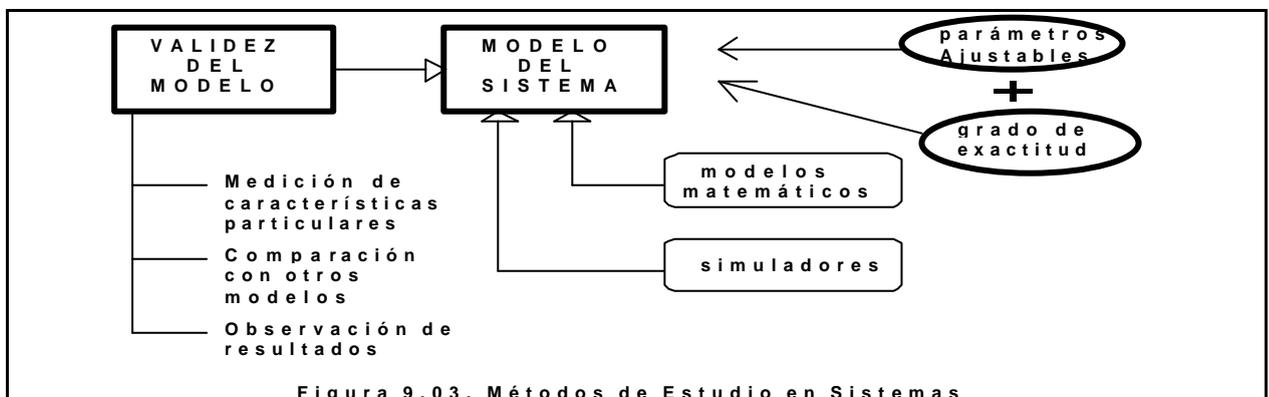


Figura 9.03. Métodos de Estudio en Sistemas

Para estudiar un sistema interesa conocer el funcionamiento del sistema real con sus programas o podemos abstraernos mediante una representación del mismo modelizándolo. En el primer caso, se puede medir el funcionamiento del hardware o medir el desempeño del software. Si se realizan las mediciones particulares de cada uno, se tendrá un panorama incompleto. En el segundo caso los modelos pueden ser estudiados mediante simuladores o mediante modelos analíticos. Si se combinan más de un método el estudio proveerá un mejor resultado.

Los métodos de estudio cuantitativo de sistemas se dividen en dos clases:

—Estudio de sistemas reales y programas:

- medidas cableadas
- medidas programadas

—Representación de sistemas y programas:

- modelos de simulación
 - modelos analíticos
- Es importante señalar que ninguno de estos métodos son suficientes por sí solos para realizar todas las tareas de evaluación. Los mejores resultados se obtienen combinando dos o más de

las técnicas mencionadas para poder dar una verdadera validez al modelo.

9.2. Modelos de Sistema

En la Figura 9.03 se plantea los elementos que intervienen en la modelización de un Sistema.

9.2.1. Objetivos de los Modelos

Básicamente se busca cubrir los siguientes objetivos:

- Construir un modelo que contenga una cierta cantidad de parámetros ajustables para estudiar el comportamiento de un sistema cualquiera.
- Construir un modelo matemático o un simulador según el grado de exactitud que se necesite.

Los modelos analíticos, si bien son muy simplificados, proporcionan fórmulas que den rápidas soluciones para todos los valores de los parámetros.

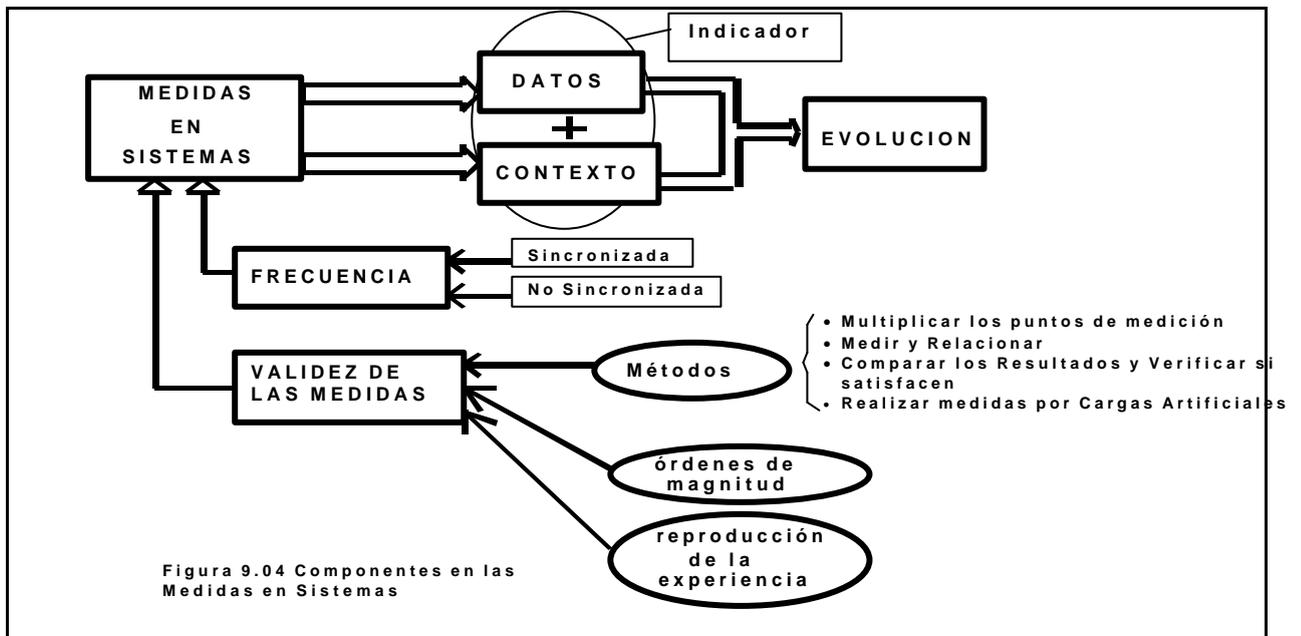
El simulador al ser más complejo permite estudiar sistemas más complejos, modificar algoritmos y utilizar leyes de probabilidad para las que se conocen pocas propiedades matemáticas. La mayor dificultad de los modelos es el comportamiento aleatorio de los sistemas. Los resultados que se obtienen son estadísticas que pueden fluctuar de tal forma que no permiten sacar conclusiones claras.

A veces es más práctico construir modelos matemáticos, que aunque sean muy simples, interesan más que un simulador elaborado. Hay que destacar que un programa de simulación de gran tamaño puede resultar costoso y difícil de implantar, lo que lo haría poco manejable para la experimentación.

9.3. Medidas sobre sistemas reales

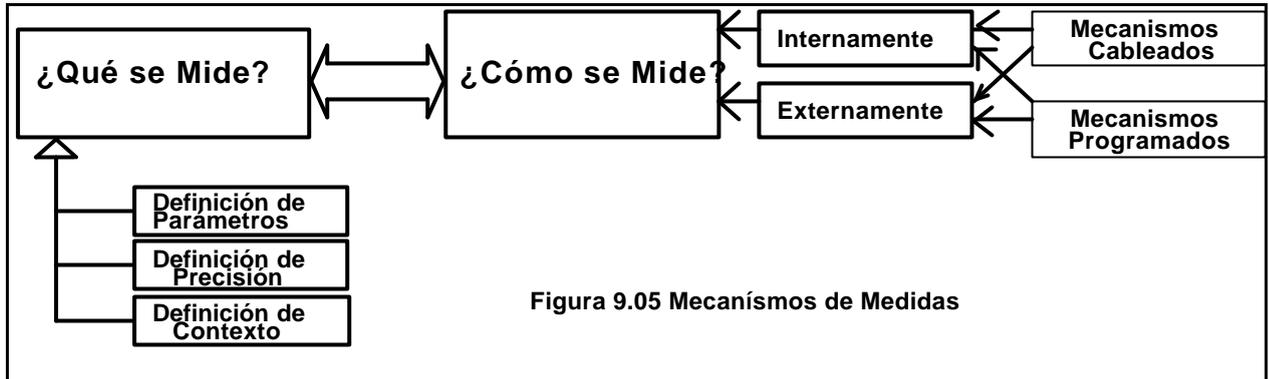
Las medidas que se hacen sobre sistemas reales, sirven, al igual que los modelos, para mejorar las prestaciones; además el sistema puede explotarlas dinámicamente, para poder controlar la carga por retroacción o por estadísticas.

Los sistemas reales permiten medir las características de las peticiones y verificar la validez de los modelos. En la Figura 9.04 proponemos los componentes que intervienen en las medidas de los Sistemas.



9.3.1. Naturaleza de las Medidas

La mayoría de los datos elementales que se recogen con mecanismos de medida son: los contenidos de memoria, los sucesos, la compactación de los sucesos (frecuencias) y estos se clasifican en los siguientes grupos.



—Clasificación de los datos:

- a) las medidas de intervalo de tiempo y las medidas instantáneas de valores que no sean tiempos (longitud de las colas de espera, volumen de memoria ocupada, longitud de los mensajes, etc.);
- b) los cálculos en el tiempo (frecuencia de los sucesos);
- c) el contenido de los conjuntos en un instante dado (colas de espera, memorias virtuales, etc.);
- d) la secuencia de sucesos que representa una sucesión de acciones en el tiempo.

— Es necesario conocer el contexto de donde se ha obtenido la cantidad. Para unir la medida a la evolución del sistema se puede:

- anotar la hora en que se hizo la medición;
- o bien referirse a sucesos que el sistema conozca.

—Se distinguen dos métodos de adquisición de datos:

- *medidas sincronizadas con los cambios de estado del sistema* (por ejemplo, mediciones periódicas);
- *medidas sincronizadas con la evolución del sistema*; en este caso se dispondrá de una información adicional que está en consonancia con la actividad del sistema.

* **¿Qué medir?** Es necesario para efectuar cualquier medida:

- ◆ Definir parámetros,
- ◆ definir precisión,
- ◆ definir contexto.

* Sobre actividades de la CPU se puede medir por ejemplo:

- ◆ IDLE : master, user
- ◆ Tiempo de ejecución
- ◆ Tiempo y frecuencia de interrupciones sobre CPU
- ◆ SYSCALL: lo cual al cambiar de modo del procesador (context switch)
- ◆ Produce OVERHEAD, el cual podría medirse
- ◆ Frecuencia de programas

* medición de retardos, bloqueos a los accesos a áreas de memoria, DMA, procesos reentrantes, E/S.

¿Cómo medir?

1. Internamente o externamente al Sistema
2. Cableadas o programadas

9.3.2. Metodología de las Medidas

—Existen varios métodos:

- **multiplicar los puntos de medida** de tal forma que se mida una misma magnitud de

- muchas formas independientes,
 - **medir independientemente magnitudes relacionadas** entre sí y verificar que se satisface dicha relación,
 - **comparar los resultados** con las precisiones de un simulador o de un modelo matemático,
 - **realizar medidas bajo una carga artificial** de características muy simples, que permita calcular resultados por adelantado.
- Durante la medición, es útil conocer el orden de magnitud, de forma que se detecten las anomalías lo antes posible.
- La reproducción de las experiencias, es un problema delicado dado que los fenómenos tienen una naturaleza aleatoria; además el sistema puede evolucionar y será necesario conocer y registrar con toda precisión, en cada experimentación que se haga, el estado de la versión que se utilice.

9.3.3. Mecanismos de Medida

Generalidades

Se pueden distinguir en mecanismos externos o internos, cableados o programados (ver Figura 9.05)

Aparatos de medidas externas

Los montajes clásicos comprenden un conjunto de sondas de alta impedancia que se pueden conectar a los circuitos de la máquina, una serie de circuitos lógicos que permitan combinar las señales de las sondas, un sistema de grabación en cinta magnética (o discos) y relojes de alta frecuencia (1 GHz) para ir anotando la hora en que se producen los sucesos que se registran. A veces el computador se encarga de tratar los datos en tiempo real.

El interés de esta técnica consiste en evitar las interferencias con el sistema que se está estudiando. Se graba una gran cantidad de información para procesarla posteriormente combinando las sondas entre ellas, por medio de circuitos electrónicos; se disminuye la cantidad de grabaciones conservando solamente la información que sea útil.

Se puede presuponer que en un futuro, las máquinas se prestarán más fácilmente a la experimentación; como por ejemplo incluir un panel de conexión, que proporcionará el fabricante, que protegerá a la máquina contra las falsas maniobras y los distintos mecanismos permitirán reunir datos sobre la utilización de los procesadores, canales y memorias. Estas medidas que se puedan elegir y fijar en un panel de comandos podrá advertir al operador de las situaciones anormales como ser, por ejemplo, sobrecargas.

Mecanismos cableados internos al sistema

El aparato de medida puede utilizar los propios recursos del sistema, como los relojes o los canales. Un montaje interesante consiste en hacer que el aparato haga el papel de un periférico conectado a un canal. En algunos casos las medidas las ejecuta un ordenador satélite, que se adapta a los problemas en tiempo real; de esta forma el sistema tiene la posibilidad de ordenar las mediciones, de leerlas para después tratarlas y hasta de utilizarlas para mejorar su propio funcionamiento.

Medidas programadas

Para implantar medidas programadas es necesario tener un conocimiento profundo del sistema. Para esto es necesario la presencia de interfaces normalizados y puntos de paso obligado, bien definidos.

Es importante tener previstas las medidas desde el momento en que se diseñe el sistema.

Las perturbaciones producidas por las propias medidas representan serias dificultades. Estas interferencias resultan difíciles de evaluar, ya que no basta con calcular la duración de la medida, debido a que la perturbación es también función de la frecuencia de dichas medidas.

Las medidas por muestreo son sencillas de implantar, aunque la información pueda no ser fácilmente accesible en todo momento. El período de muestreo no debe estar sincronizado con el comportamiento dinámico del sistema que se estudie, sino se deformarán las medidas. Además una coincidencia tal es poco probable que se dé en los sistemas complejos cuyo comportamiento sea muy variable.

9.3.4. Utilización de las Medidas

Evaluación de los sistemas

La Figura 9.06 presenta todos los componentes que intervienen en la evaluación del sistema integro (Software + Hardware + ambiente sin descartar la influencia de las personas, que por ejemplo, planifican la carga de trabajo del sistema u operan el mismo).

El problema de la evaluación de los sistemas se puede ver desde dos puntos de vista:

- definición de las magnitudes que caracterizan a las prestaciones
 - la especificación de las características de la carga del sistema.
- Cualquier estimación que se haga de las prestaciones debe ir referida a una carga bien definida.
 - El funcionamiento global de un sistema se caracteriza por dos dimensiones:
 - la velocidad de los trabajos
 - las tasas de utilización de los distintos recursos, la cual no depende solamente de la petición, sino también de la gestión que haga el Kernel del conjunto de los recursos.
 - La carga real de un sistema es variante y no reproducible. Esto lleva a definir cargas artificiales que satisfacen las siguientes condiciones:
 - ser representativas de las condiciones de explotación reales del sistema que se va evaluar, en el sentido que se precise.
 - ser reproducibles.
 - ser fáciles de parametrizar.
 - Al evaluar el sistema es importante tener en cuenta: **Software + Hardware + contexto**

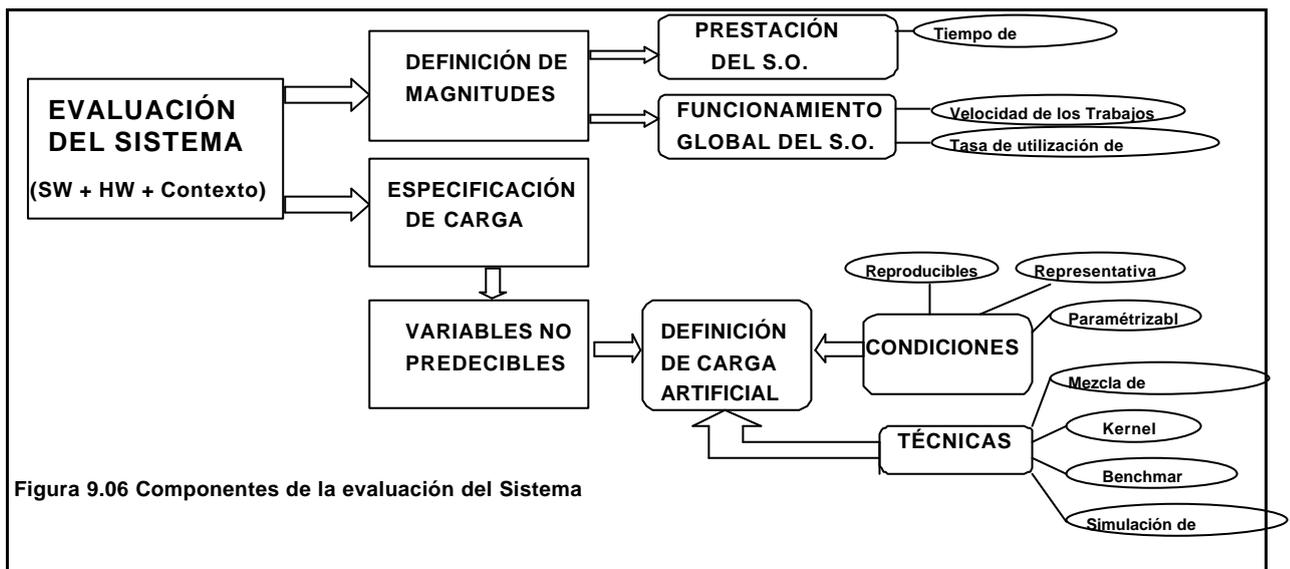


Figura 9.06 Componentes de la evaluación del Sistema

9.4. Tendencias importantes que afectan los aspectos de desempeño

En los primeros años del desarrollo de los sistemas de cómputo, el equipo representaba el costo dominante, de manera que los estudios sobre desempeño se concentraban ante todo en aspectos del equipo. Actualmente los equipos son relativamente económicos y su costo baja cada vez más. Sin embargo, los costos de la mano de obra han seguido creciendo en forma constante. En consecuencia, debemos comenzar a medir el desempeño con técnicas más acordes con la productividad humana. Por ejemplo, un computador personal sirve primordialmente para dar apoyo a un individuo, de modo que el porcentaje de utilización del procesador y de los dispositivos de E/S tiene menor importancia que la preparación del computador para responder con rapidez en el momento en que lo necesite el usuario.

La complejidad de los programas va en aumento con la generalización del empleo de sistemas operativos de multiprogramación / multiprocesamiento, sistemas distribuidos, sistemas para administración de bases de datos, interfases con el usuario orientadas a los gráficos y diversos sistemas de apoyo a aplicaciones. Los programas ocultan a menudo el equipo al usuario, con lo que crean una máquina virtual definida por las características de operación de los programas. Con frecuencia los programas de difícil manejo provocan un desempeño pobre, aun en sistemas con equipo poderoso, por lo que es importante examinar el desempeño de los programas de un sistema además del desempeño de su equipo.

9.5. Por qué es necesario supervisar y evaluar el desempeño

Tres objetivos comunes de la evaluación del rendimiento:

1. *Evaluación para selección*: El evaluador del desempeño debe decidir si conviene adquirir un sistema de cómputo de un proveedor específico.
2. *Proyección del rendimiento*: El objetivo del evaluador en este caso es estimar el desempeño de un sistema inexistente. Puede tratarse de un sistema de cómputo totalmente nuevo o de un nuevo componente de hardware o software.
3. *Supervisión del desempeño*: El evaluador acumula datos del rendimiento de un sistema o componente ya existente para asegurarse de que cumple con sus objetivos de desempeño, para ayudar a estimar el impacto de las modificaciones planeadas y para ofrecer a los administradores la información que necesitan para tomar decisiones estratégicas tales como modificar o no un sistema existente de prioridades de trabajo.

La evaluación y predicción del desempeño se necesitan durante:

- Los primeros momentos de la concepción y desarrollo de un sistema nuevo: el proveedor intenta predecir la naturaleza de las aplicaciones que se ejecutarán en el sistema, así como las cargas de trabajo que deben manejar dichas aplicaciones.
- En el desarrollo y realización del nuevo sistema sirven para determinar la mejor organización del equipo, las estrategias de administración de recursos que deben ponerse en práctica en el sistema operativo y si el sistema en evolución cumple o no con sus objetivos de rendimiento.
- En el lanzamiento al mercado el proveedor debe estar preparado para responder a preguntas de usuarios potenciales acerca de la capacidad del sistema para manejar ciertas aplicaciones con ciertos niveles de rendimiento.
- En la instalación tanto el usuario como el proveedor están interesados en obtener un desempeño óptimo. Se realizan ajustes finos para que el sistema funcione de la mejor manera posible dentro del ambiente de operación del usuario (proceso de *afinación del sistema*).

9.6. Medidas de desempeño

Desempeño (o rendimiento) es la manera en la que un sistema de cómputo cumple con sus objetivos, o la eficiencia con la que lo hace. Es una cantidad relativa más que absoluta, aunque se habla de *medidas absolutas de desempeño* tales como el número de trabajos por hora que puede atender un sistema de cómputo dado. Siempre que se estima una medida del desempeño, lo normal es que sirva como base para comparaciones.

Tipos de medidas del desempeño:

- *Diffíciles de cuantificar*: tales como la facilidad de uso.
- *Fáciles de cuantificar*: tales como el número de accesos a disco por minuto.
- *Orientadas al usuario*: como el tiempo de respuesta
- *Orientadas al sistema*: como el porcentaje de utilización del procesador.
- Algunas de las medidas comunes del desempeño son:
- *Tiempo de retorno (turnaround time)*: En un sistema de procesamiento por lotes, se define como el tiempo transcurrido desde la entrega de un trabajo hasta su devolución al usuario.
- *Tiempo de respuesta (Response Time)*: Es el tiempo de retorno en un sistema interactivo, y a menudo se define como el tiempo transcurrido desde que el usuario presiona una tecla

Enter (Intro) u oprime el botón de un ratón hasta que el sistema comienza a imprimir o exhibir una respuesta.

- **Tiempo de reacción del sistema:** En un sistema interactivo, se define a menudo como el tiempo transcurrido desde que el usuario presiona una tecla *Enter (Intro)* u oprime el botón de un ratón hasta que se otorga la primera porción de tiempo de servicio a esa solicitud del usuario.

Estas son cantidades probabilísticas, y en los estudios de simulación y modelado de sistemas se consideran como variables aleatorias.

Cuando se habla del *valor esperado* de una variable aleatoria, se refiere a un *valor promedio o medio*. Pero los promedios muchas veces pueden ser engañosos porque un valor medio dado se puede producir promediando una serie de valores idénticos o casi idénticos, o bien promediando una amplia variedad de valores, algunos muchos más grandes y algunos muchos más pequeños que el promedio calculado. Por eso se utilizan a menudo otras medidas del desempeño:

- **Varianza de los tiempos de respuesta** (o de cualquiera de las variables aleatorias mencionadas): La varianza de los tiempos de respuesta es aun medida de la *dispersión*. Una varianza pequeña indica que los tiempos de respuesta que perciben los usuarios están en general cerca de la media. Una varianza considerable indica que algunos usuarios pueden estar recibiendo tiempos de respuesta muy distintos de la media. La varianza de los tiempos de respuesta es una medida de su predecibilidad. Desde el punto de vista del factor humano, ésta puede ser una medida muy importante del desempeño de un sistema interactivo.
- **Producción:** Es la medida de rendimiento de trabajo por unidad de tiempo.
- **Carga de trabajo:** Es la medida de la cantidad de trabajo que se ha introducido en el sistema, cantidad que el sistema debe procesar en condiciones normales para que su funcionamiento se considere aceptable.
- **Capacidad:** Es una medida de la producción máxima que puede tener un sistema, si suponemos que cada vez que el sistema está preparado para aceptar más trabajos, hay otro trabajo inmediatamente disponible.
- **Utilización:** Es la fracción de tiempo que está en uso algún recurso. La utilización puede ser una medida engañosa. Aunque al parecer lo mejor es tener un porcentaje de utilización alto, esto puede ser el resultado de un aprovechamiento ineficiente. Una forma de lograr un alto porcentaje de utilización del procesador, por ejemplo, es ejecutar una serie de procesos, cada uno de los cuales está en un ciclo infinito. Otro punto de vista sobre la utilización del procesador ofrece también observaciones interesantes. Se puede considerar que el procesador en un momento dado está ocioso, en estado de programa o en estado de supervisión. Cuando un procesador está en estado de programa realiza labores en beneficio de un usuario, y lo más probable es que el costo se cobre a dicho usuario. Sin embargo, cuando el procesador está en el estado de supervisión, se encuentra dentro del sistema operativo. Parte de esta tiempo se puede cobrara directamente a los usuarios, pero una parte, como el tiempo ocupado en conmutar entre diversos procesos en los sistemas de programación, no es sino un gasto general. Este componente de gasto general puede ser considerable en algunos sistemas. Así, cuando medimos la utilización del procesador, debe interesarnos cuánto de este uso es trabajo productivo en beneficio de los usuarios, y cuánto es un gasto general del sistema.

9.7. Técnicas para evaluar el desempeño

A continuación daremos una breve explicación de las distintas técnicas que se usan para evaluar el desempeño de un sistema de cómputo

9.7.1. Tiempos

—Los *tiempos* permiten hacer comparaciones rápidas entre equipos de cómputo.

Los primeros computadores se evaluaban muchas veces en términos de sus tiempos de suma o de sus tiempos de ciclo de memoria. Los tiempos son útiles para indicar el “caballaje bruto” de un sistema de cómputo específico y se ha hecho común caracterizar un computador dado por el número de MIPS (millones de instrucciones por segundo) ejecutadas. Existen computadores con capacidad de ejecución de

BIPS (miles de millones de instrucciones por segundo), y es posible que aparezcan máquinas que trabajen en el intervalo TIPS (billones de instrucciones por segundo).

- Ventajas:

- Con la llegada de las *familias de computadores* se ofrecen computadores con los cuales un usuario puede progresar a procesadores más rápidos conforme a sus necesidades. Los tiempos ofrecen una forma rápida de comparar entre sí los miembros de una familia de computadores.
- Los tiempos se usan sobre todo para comparaciones rápidas.

- Desventajas:

- Se requieren otras técnicas para evaluar en forma más significativa los complejos sistemas de hardware y software de la actualidad. En general sólo se hacen comparaciones de tiempos para unas cuantas operaciones básicas del equipo. Por tanto, al comparar procesadores para una aplicación específica, los tiempos ofrecen muy poca información.

La controversia sobre RISC vs. CISC indica que los tiempos deben interpretarse con cuidado en el contexto de las deferencias fundamentales en la arquitectura.

9.7.2. Mezcla de Instrucciones (Instruction Mix)

La técnica de *mezclas de instrucciones* emplea un promedio ponderado de diversos tiempos de instrucciones más adecuado para una aplicación específica.

El evaluador del desempeño estudia la mezcla de trabajos en una instalación específica e intenta formar un promedio ponderado de tiempos para aquellas instrucciones de uso más frecuente en la instalación.

Es una simulación de la carga donde las instrucciones de varios tipos (aritméticas, de comparación, de transferencia) figuran con una determinada frecuencia; ésta se puede obtener de las medidas que se hayan hecho en una carga real. Se intenta así poder caracterizar un tipo dado de aplicación. Si se trata de una mezcla sencilla, el tiempo de ejecución puede calcularse manualmente.

Se han propuesto distintas "Mezclas", representativas de cargas de trabajo en un sistema, que se emplean para establecer los **Tiempos de ejecución promedio** para una CPU:

$$tE = \sum_{i=1}^n P_i * t_i$$

Para:

tE = Tiempo de Ejecución promedio

P_i = Probabilidad de ocurrencia de una instrucción del tipo i

t_i = tiempo de ejecución promedio de una instrucción del tipo

Esta medida ofrece un promedio de todas las instrucciones de CPU ajustadas con su frecuencia de ejecución de las instrucciones, pues:

$$f_E = 1/tE \text{ (Frecuencia de Ejecución de Instrucciones).}$$

- Ventajas:

- Las comparaciones entre máquinas pueden tener más validez que si se usan los tiempos por sí solos.
- Las mezclas son útiles en su mayor parte para comparaciones rápidas.

- Desventajas:

- El evaluador debe advertir que conviene utilizar otras técnicas antes de tomar decisiones de adquisición vitales. Las ponderaciones empleadas en esta técnica tienden a ser muy subjetivas, por lo que es preciso examinar con cuidado los puntos finos de los conjuntos de instrucciones de diferentes sistemas de cómputo.
- Las mezclas ofrecen muy poca o ninguna información útil para evaluar programas.
- La complejidad del diseño de los equipos actuales hace cada vez más difícil establecer la validez de las mezclas de instrucciones. Con el empleo de memoria caché y canalización, una misma instrucción puede ejecutarse en tiempos diferentes, dependiendo de su contexto en las distintas ejecuciones.

Los estudios que determinan mezclas de instrucciones de diversas índoles siguen gozando de popularidad, sobre todo en la evaluación de enfoques arquitectónicos RISC vs. CISC.

9.7.3. Programas Núcleos (Kernels)

—Es una ampliación del sistema anterior pero utilizando un conjunto de sub-programas.

—Los métodos de mezcla de instrucciones y los Kernels se utilizan sobre todo para comparar entre sí las prestaciones de unidades centrales diferentes; como no llevan operaciones de entrada - salida, no sirven para evaluar sistemas.

Tanto los tiempos como las mezclas tienen el defecto de que destacan sólo unos cuantos aspectos del conjunto de instrucciones de un computador. Un *programa núcleo* es un programa representativo que podría ejecutarse en una instalación. Los tiempos del programa núcleo para una máquina dada se miden empleando las estimaciones de tiempos de las instrucciones del fabricante. Las comparaciones entre máquinas se realizan con base en las diferencias en los tiempos de ejecución esperados del programa núcleo en las distintas máquinas. De este modo el núcleo “se ejecuta en papel” en vez de ejecutarse en un computador. (Nota: No debe confundirse el término “núcleo” empleado con el núcleo (kernel) del sistema operativo).

—Ventajas:

- ◇ Los núcleos producen mejores resultados que los tiempos o las mezclas de instrucciones.
- ◇ Una ventaja decisiva es que los núcleos son programas completos y , en última instancia, esto es lo que el usuario ejecuta en realidad es el sistema de cómputo considerado.
- ◇ Los núcleos pueden ser útiles para evaluar ciertos componentes de software de un sistema; por ejemplo, dos compiladores distintos podrían producir códigos drásticamente diferentes y los núcleos podrían ayudar a decidir cuál compilador produce el código más eficiente.

—Desventajas:

- ◇ La preparación y cronometraje de los programas núcleos requieren un considerable esfuerzo manual.

9.7.4. Modelos Analíticos

Los *modelos analíticos* son representaciones matemáticas de sistemas de cómputo o de sus componentes. Se utilizan muchos tipos de modelos, pero los de la *teoría de colas* y los *procesos de Markov* son al parecer los más manejables y útiles.

—Ventajas:

- Para los evaluadores con inclinaciones matemáticas, el modelo analítico puede resultar fácil de crear y modificar y ser una gran ayuda para estimar con rapidez y relativa precisión el desempeño de un sistema o componente de cómputo.

—Desventajas:

- los evaluadores deben ser expertos matemáticos y éstos no abundan en los ambientes de cómputo comercial
- sólo existen soluciones claras para los modelos más sencillos
- puede llevar a realizar suposiciones simplificadoras y destruir la utilidad y aplicabilidad del modelo

El evaluador de rendimiento debe entender muchas técnicas diferentes y debe saber utilizarlas en forma combinada y lograr que las evaluaciones se refuercen entre sí.

9.7.5. Juegos de Ensayo y Comparación (Benchmark)

- Es un conjunto de programas que se consideran representativos de algún tipo de carga.
- Pueden comprender programas reales o los que se han creado especialmente para esta prueba.
- Según sea su composición, pueden utilizarse para probar todas las partes del sistema, incluso

los programas de servicio (compiladores) y las entradas-salidas.

Un *banco de prueba* es un programa real que el evaluador ejecuta en la máquina que está evaluando. Lo más común es que el banco de prueba sea un *programa de producción* (es decir, uno ejecutado con regularidad en la instalación) representativo de muchos trabajos en dicha instalación.

El evaluador conoce muy bien el desempeño del banco de prueba en el equipo existente, de modo que cuando lo ejecuta en un nuevo equipo puede sacar conclusiones significativas.

- Ventajas:
 - El evaluador no tendrá que crear el banco de prueba, sólo necesita escogerlos de entre los programas de producción conocidos.
 - No se realizan cronometrajes manuales de instrucciones individuales. El programa completo se ejecuta con datos reales en la máquina real, de modo que el computador realiza la mayor parte del trabajo.
 - Se reducen al mínimo las posibilidades de error humano.
 - El tiempo se puede medir con un cronómetro o lo puede medir el computador mismo.
 - Los bancos de prueba resultan valiosos en ambientes con sistemas de multiprogramación, tiempo compartido, multiprocesamiento, bases de datos, comunicaciones de datos, y tiempo real porque se ejecutan en la máquina real en condiciones reales.
 - Útiles para evaluar programas, aun en ambientes de operación complejos.
 - Son de utilidad para comparar el funcionamiento de un sistema antes y después de realizar modificaciones.
- Desventajas:
 - Selección cuidadosa de una serie de bancos de prueba representativa de las características de trabajo en la instalación (proceso subjetivo).
 - No resultan útiles para predecir los efectos de cambios propuestos, a menos que exista otro sistema en el que ya se hayan incorporado las modificaciones y en el cual puedan ejecutarse los bancos de prueba.

Es la técnica más usada entre las empresas establecidas que están estudiando la adquisición de equipo de varios proveedores distintos.

9.7.6. Programas Sintéticos

Los *programas sintéticos* combinan las técnicas de los núcleos y los bancos de prueba. Son programas reales que se han diseñado a la medida para ejercitar funciones específicas de una máquina.

- Ventajas:
 - Podrían no existir programas de banco de prueba para probar funciones de una nueva máquina.
 - Útiles en ambiente de desarrollo: al quedar disponibles nuevas funciones, pueden usarse programas sintéticos para probar si trabajan bien.
- Desventajas:
 - Los evaluadores no siempre tienen suficiente tiempo para codificar y depurar programas sintéticos.

Programas sintéticos estándar muy usados: Whetstone y Dhystone.

9.7.7. Simuladores de Carga

Es un programa que utiliza de forma continua y parametrizada los distintos recursos de los sistemas. De este modo, se pueden estudiar los efectos que tienen sobre el sistema, los distintos parámetros de la carga.

En el caso de los sistemas conversacionales de acceso múltiple, el simulador de carga deberá simular además el comportamiento de los distintos usuarios; para ello deberá incorporar o utilizar un dispositivo de gestión de procesos paralelos. La dificultad se encuentra en separar dentro de las mediciones el efecto de la carga simulada y el de la gestión interna del simulador que utiliza también los

recursos del sistema. Esta dificultad se puede solucionar permitiendo que se ejecute el programa de simulación en otro ordenador que esté unido al sistema. Esta técnica se adapta bien a la simulación de los usuarios conversacionales y en general, a todos los fenómenos que dependan del tiempo real.

La *simulación* es una técnica en la que el evaluador desarrolla un modelo computarizado del sistema que desea evaluar. El modelo se ejecuta a continuación de modo que refleje el comportamiento del sistema en evaluación.

- Tipos de simuladores:
 - *Simuladores manejados por los eventos:* Son controlados por los eventos producidos en el simulador de acuerdo con distribuciones de probabilidades.
 - *Simuladores manejados por libreto:* Son controlados por datos derivados empíricamente y manipulados con cuidado de modo que reflejen el comportamiento esperado del sistema simulado.
- Ventajas:
 - Es posible preparar un modelo de un sistema que no exista y después ejecutar el modelo para ver cómo se comportaría el sistema en ciertas circunstancias.
 - Pueden evitar la construcción de sistemas mal diseñados poniendo en relieve los problemas en una etapa temprana del ciclo de desarrollo.
 - Luego de *establecer la validez* de los modelos de simulación (es decir, que son en verdad representaciones precisas de los sistemas reales simulados), su uso repetido puede resultar provechoso y económico.
- Desventajas:
 - Requiere evaluadores con un alto nivel de conocimientos y puede consumir bastante tiempo de cómputo.
 - Producen cantidades tremendas de datos que se deben analizar manualmente o bien por computador.

Han adquirido popularidad en las industrias espacial y de transporte, debido a la seriedad de las consecuencias que tendría la construcción de un sistema que fallara.

9.7.8. Supervisión del Desempeño

La *supervisión del desempeño* es la obtención y análisis de información acerca del rendimiento de sistemas ya existentes.

Puede realizarse usando técnicas de software o de hardware.

- Ventajas:
 - Resulta útil para averiguar lo bien que se comporta un sistema en términos de producción, tiempos de respuesta, predecibilidad, etc.
 - Es posible localizar rápidamente los cuellos de botella y ayudar a los administradores a decidir cómo puede mejorarse el rendimiento.
 - Los supervisores indican con precisión cómo está funcionando el sistema, y la gran cantidad de información producida puede ser en extrema valiosa, sobre todo en ambientes de desarrollo en los cuales quizá deban tomarse o modificarse decisiones clave de diseño con base en la operación observada del sistema.
 - Los *rastros de ejecución de instrucciones* o los *rastros de ejecución de módulos* muchas veces pueden poner al descubierto *cuellos de botella*. Un rastreo de ejecución de módulos podría mostrar que un pequeño subconjunto de los módulos se está usando gran parte del tiempo. Por lo tanto, si los diseñadores concentran sus esfuerzos de optimización en esos módulos, pueden mejorar en forma considerable el desempeño del sistema sin invertir trabajo y recursos en partes del sistema de uso poco frecuente.
- Desventajas:
 - Los programas supervisores pueden distorsionar las lecturas de rendimiento porque ellos mismos consumen recursos del sistema.
 - Los equipos supervisores son en general más costosos, pero con mínima influencia sobre la operación del sistema.

- Los supervisores producen enormes volúmenes de datos que deben analizarse, lo cual requiere considerables recursos de cómputo.

Mejora de las prestaciones

Si se sabe a que partes del programa se dedica la mayoría del tiempo de ejecución, se podrán encaminar los esfuerzos para optimizar dichas partes; si el programa se ejecuta en una memoria paginada, las medidas pueden detectar una dispersión de las referencias a las páginas, cuya corrección ayudará a mejorar las prestaciones del programa.

Otra aplicación afecta a la mejora de las prestaciones de un sistema operativo, en el que se podrán detectar cuellos de botella haciendo mediciones. Siempre se puede mejorar... es una función directa de tiempo, costos y esfuerzos.

9.7.9. Whetstone, Dhrystone, Linpack y Savage

Los usuarios pueden crear programas sintéticos y bancos de prueba a la medida de sus necesidades, o pueden utilizar ciertos estándares de amplia aceptación.

- *Whetstone* es uno de los más antiguos de los estándares de mayor publicación; es un programa sintético de banco de prueba diseñado para medir la efectividad de un procesador que realiza grandes cantidades de cálculos de punto flotante. Aunque se está volviendo algo obsoleto, sigue siendo un método útil para comparara la efectividad con la cual los procesadores ejecutan programas en FORTRAN para cálculos científicos.
- Los programas *LINPACK* (*SLINPACK* para cálculos de precisión sencilla y *DLINPACK* para cálculos de doble precisión), desarrollados en el Argonne National Laboratory, se usan mucho también para medir el desempeño en operaciones de punto flotante. *LINPACK* es un programa de banco de prueba genuino; no es sintético como *Whetstone*.
- El banco de prueba *Savage*, a diferencia de los programas *Whetstone* y *LINPACK*, mide la precisión de las operaciones trigonométricas de punto flotante, más que su rendimiento.
- *Dhrystone* es un banco de prueba sintético desarrollado para medir la efectividad con la cual una arquitectura ejecuta programas de sistemas. Las versiones para Ada, C y Pascal son de uso común. Los creadores de *Dhrystone* hacen hincapié en que las variaciones en las arquitecturas, sobre todo el empleo de memoria caché, pueden afectar de manera significativa la velocidad de ejecución del programa *Dhrystone*. Como el programa es pequeño, su efectividad es especialmente sensible al tamaño de la memoria caché; si el programa cabe en ella, se ejecutará mucho más rápido que si tiene que estar transfiriéndose entre la memoria caché y la principal.
- Los lenguajes, compiladores, métodos de medición y el empleo de memoria caché son todos factores que deben tomarse en cuenta para que tengan sentido las comparaciones basadas en la medición del desempeño.

9.8. Cuellos de botella y saturación

El sistema operativo administra un conjunto de recursos, los cuales se comunican e interactúan en formas complejas para llevar a cabo la operación total del sistema. En ocasiones, uno o más de estos recursos se convierten en *cuellos de botella*, con lo que limitan el desempeño total del sistema al no ser capaces de realizar su parte del trabajo. En tanto que otros recursos del sistema pueden tener capacidad de sobra, es probable que los cuellos de botella no estén enviando trabajos o procesos a estos otros recursos con la rapidez necesaria para mantenerlos ocupados.

Un recurso tiende a convertirse en un cuello de botella cuando el tráfico de trabajos o procesos en ese recurso se aproxima a su capacidad. En este momento se dice que el recurso está *saturado*; es decir, los procesos que compitan por la atención del recurso comienzan a interferir unos con otros. Ejemplo: hiperpaginación.

Detección de cuellos de botellas: deben supervisarse las colas de solicitudes de cada recurso; cuando una cola comienza acrecer con rapidez es porque la *tasa de llegada* de solicitudes a ese recurso es mayor que su *tasa de servicio*, de modo que el recurso se ha saturado.

El aislamiento de los cuellos de botella es una parte importante de la afinación de un sistema. Los cuellos de botella pueden eliminarse aumentando la capacidad de los recursos o añadiendo más recursos de ese tipo en ese punto del sistema. No obstante, la eliminación de un cuello de botella no siempre mejora la producción, pues es posible que existan otros cuellos de botella en el sistema. La afinación de un sistema puede requerir repetidas localizaciones y eliminaciones de cuellos de botella hasta que el desempeño del sistema alcance niveles satisfactorios.

9.9. CICLOS DE RETROALIMENTACIÓN

Un *ciclo de retroalimentación* es un caso en el que cierta información acerca del estado actual del sistema puede afectar las solicitudes que llegan. Si la retroalimentación indica que dichas solicitudes pueden tener problemas para ser atendidas, quizá se les pueda evitar por otro conducto. La retroalimentación puede ser negativa, en cuyo caso pueden disminuir las tasas de llegada, o positiva, en cuyo caso pueden aumentar.

9.9.1. Retroalimentación Negativa

En situaciones de *retroalimentación negativa*, la tasa de llegada de las solicitudes nuevas puede disminuir como resultado de la retroalimentación de información.

En sistemas distribuidos, las salidas por *spool* a menudo pueden ser impresas por cualquiera de varios despachadores de impresora equivalentes. Si la cola de una impresora es demasiado larga, el trabajo puede colocarse en otra cola menos congestionada.

La retroalimentación negativa contribuye a la *estabilidad* en los sistemas de colas. Por ejemplo, si al llegar los trabajos se colocan indiscriminadamente en cola para un dispositivo muy ocupado, la cola de ese dispositivo podría crecer en forma indefinida (aun cuando otros dispositivos pudieran estar subutilizados).

9.9.2. Retroalimentación Positiva

En situaciones de *retroalimentación positiva*, la tasa de llegada de nuevas solicitudes puede aumentar a causa de la retroalimentación de información.

Desventajas: Un ejemplo clásico de los problemas que pueden surgir cuando hay retroalimentación positiva se presenta en sistemas de multiprogramación con memoria virtual paginada. Supongamos que el sistema operativo detecta que el procesador está subutilizado. Podría indicarle al planificador de trabajo que admita más trabajos en la mezcla de multiprogramación, con la intención de que ello aumente la carga sobre el procesador; al admitirse más trabajos, disminuye la cantidad de memoria que o puede asignarse a cada uno y quizás aumente la cantidad de fallas de página (porque los conjuntos de trabajo no caben en la memoria). El porcentaje de utilización del procesador puede disminuir por la hiperpaginación en el sistema. Un sistema operativo de diseño deficiente podría decidir admitir todavía más trabajos y, desde luego, tal cosa provocaría un mayor deterioro en la utilización del procesador.

Los diseñadores de sistemas operativos deben tener cuidado al diseñar mecanismos con retroalimentación positiva para evitar que se presente una operación inestable como la del ejemplo. Es preciso supervisar los efectos de cada modificación incremental para ver si produce la mejora esperada. Si una modificación hace que se deteriore el rendimiento, esto indicará al sistema operativo que puede estar entrando en un intervalo inestable, de modo que se hace necesario ajustar las estrategias de asignación de recursos para continuar con una operación estable.

9.10. COPROCESADORES

Los *coprocesadores* son procesadores de aplicación especial, con funciones que quizá no tenga el procesador o que quizá tenga pero no realice bien.

Los coprocesadores más comunes que hay ahora en el mercado son *coprocesadores de punto flotante*, los cuales ejecutan operaciones comunes de punto flotante a muy altas velocidades en hardware; sin ellos dichas operaciones se ejecutan por lo regular en software (con relativa lentitud). Existen coprocesadores para facilitar los gráficos de alta velocidad, la síntesis y el reconocimiento del lenguaje hablado y la visión por computador.

Los coprocesadores casi siempre funcionan como *procesadores esclavos* del procesador o procesadores centrales; el procesador principal asigna cálculos apropiados al coprocesador y recupera de inmediato el control cuando el coprocesador termina su trabajo.

Los coprocesadores de aplicación especial funcionan a veces como *procesadores iguales* con respecto a los procesadores principales; esto significa que los coprocesadores pueden trabajar en paralelo con los procesadores principales o unos con otros.

El ancho de banda entre la CPU y los coprocesadores puede convertirse en un cuello de botella. Algunos coprocesadores de punto flotante se manejan como parte de la memoria para ayudar a aliviar el problema.

9.11. Computación con un conjunto reducido de instrucciones (RISC)

La *computación con un conjunto reducido de instrucciones (RISC, reduced instruction set computing)* se ha popularizado como una forma de realización de uniprocadores de alto rendimiento.

La tendencia obvia a principio de los ochenta era hacia la incorporación de secciones de código de uso frecuente en instrucciones individuales de lenguaje de máquina, con la esperanza de lograr que estas funciones se ejecutaran más rápido. Este enfoque recibe el nombre de *CISC (complex instruction set computing; computación con un conjunto complejo de instrucciones)*, en respuesta a la popularidad del término RISC.

El empleo de instrucciones más complejas tienen origen en la época en que la programación de sistemas se realizaba en lenguaje ensamblador; los programadores podían escribir sus programas con menos líneas de código y aumentaban la productividad. La tendencia se generalizó con el uso de lenguajes de alto nivel para escribir sistemas operativos (tales como C, C++ para escribir sistemas UNIX).

Los diseñadores de computadores de alto rendimiento han rechazado las instrucciones complejas. Estudios indican que casi todos los programas generados por los compiladores de uso más común utilizaban sólo una pequeña porción de los conjuntos de instrucciones. Las instrucciones mejoradas se utilizaban con poca frecuencia. La interpretación mediante microcódigo de las instrucciones complejas estaba de hecho haciendo más lenta la ejecución de instrucciones sencillas.

• **Ventajas de RISC sobre CISC:**

- Las arquitecturas RISC tienden a ser “ricas en registros”, el elevado número de registros se utiliza de diversas maneras para mejorar el rendimiento. Los compiladores RISC están diseñados para optimizar el código de modo que se ejecute eficientemente con arquitecturas canalizadas.
- Muchas instrucciones de lenguaje de máquina en arquitecturas CISC requieren varios ciclos de máquina, en las RISC las instrucciones casi siempre requieren un solo ciclo de máquina.
- Se evita la microprogramación en favor del control “integrado de circuitos”. El conjunto de instrucciones es reducido y sencillo; se usan muchos registros de alta velocidad para realizar cálculos con rapidez; el tráfico entre la memoria y los registros se maneja con cargas y almacenamientos sencillos; se diseñan compiladores optimizadores que trabajen de manera efectiva con las instrucciones más sencillas y con los conjuntos abundantes de registros; se utiliza un mínimo de modos de especificar direcciones.
- Al ofrecer muchos registros, las máquinas RISC permiten a los compiladores optimizadores generar código que deje resultados intermedios en algunos registros, evitando gran parte del tráfico entre los registros y la memoria que se realiza en máquinas con relativamente pocos registros. La ejecución canalizada en arquitecturas CISC puede hacerse lenta por los tiempos de procesamiento más largos de las instrucciones complejas.
- Se utilizar una técnica denominada *ramificación retardada*. Cuando se ejecuta una ramificación condicional, podría o no ejecutarse enseguida la próxima instrucción en la secuencia, dependiendo de la evaluación de la condición. La ramificación retardada permite a esta siguiente instrucción ejecutarse de todos modos; después si no se ramifica, la siguiente instrucción puede estar ya bastante adelantada, si no terminada.
- Al ser más sencillos, los conjuntos de instrucciones RISC facilitan la tarea del compilador optimizador de detectar y aprovechar las oportunidades para optimización. Tienden a usar diversas memorias caché para agilizar la ejecución.
- Los programas RISC se ejecutan en general dos o más veces más rápido que sus equivalentes CISC.

- Una ventaja de los diseños RISC incluida en los chips VLSI es que su sencillez facilita el *cambio de escala*; es decir, los diseños RISC se pueden modificar fácilmente para ejecutarse con mayor rapidez empleando tecnologías nuevas que meten números cada vez mayores de elementos de circuito en los chips.
- Las instrucciones RISC ocupan palabras de memoria con longitud fija; tales instrucciones se ejecutan con mayor facilidad y rapidez que las de longitud variable.
- Las instrucciones RISC casi siempre se ejecutan en cuatro fases: obtención de la instrucción, lectura de registro, operación aritmética/lógica y escritura de registro.
- **Desventajas:**
 - Aunque las arquitecturas RISC reducen el tráfico entre los registros y la memoria, siguen siendo necesarias las cargas y los almacenamientos, los cuales requieren tiempo. En consecuencia, los compiladores optimizadores reacomodan las secuencias de instrucciones para que puedan realizarse otras operaciones mientras tiene lugar el tráfico entre los registros y la memoria.
 - Los programas escritos para máquinas RISC suelen tener más instrucciones por programa que los de las máquinas CISC.
 - La sencillez de las instrucciones, el conjunto reducido de instrucciones y el enfoque de equipo rico en registros forzaron a un aumento considerable en la complejidad del *cambio de contexto* (el cual debe ser *rápido*). Las máquinas RISC utilizan un número de registros muchas veces mayor que las máquinas CISC, y estos registros se deben conservar. Utilizan, también, conductos profundos que deben purgarse durante el cambio de contexto, lo que podría hacer que se perdiera gran parte de lo ganado en cuestión de rendimiento por la canalización.
 - Las operaciones de punto flotante se ejecutan con mayor rapidez en las arquitecturas CISC, de modo que los usuarios de RISC que necesitan operaciones de punto flotante rápidas añaden a menudo coprocesadores de punto flotante a sus equipos.
- **Tendencias:**
 - Un fenómeno interesante es la posibilidad de escoger subconjuntos RISC de máquinas CISC, de modo que quizá sea posible mejorar el rendimiento aun en máquinas CISC que emplean diversas técnicas RISC, como por ejemplo en la optimización del compilador.
 - La tecnología RISC ha tenido otro impacto interesante sobre las comunidades dedicadas a las arquitecturas de computadores y a los sistemas operativos. Es posible llevar a la práctica arquitecturas más sencillas con mayor rapidez y con menos recursos. Hoy día ya no son aceptables, los largos ciclos de desarrollo que se requerían antes para introducir nuevas arquitecturas importantes, porque la tecnología está avanzando con demasiada velocidad. Las suposiciones básicas y los principios fundamentales pueden cambiar tan radicalmente durante un proyecto que un sistema nuevo bien podría hacerse obsoleto antes de lanzarlo al mercado. La naturaleza total de la industria de cómputo está cambiando, a la vez que continúa en aumento el atractivo de la sencillez.
 - Conclusiones de Davidson y Vaughan que emplearon la técnica de *subconjunto arquitectónico* para realizar un estudio sobre los efectos de la complejidad de las instrucciones sobre el desempeño (eligieron tres subconjuntos de instrucciones VAX, cada uno más complejo que el anterior):
 - 1) Los programas escritos en el más sencillo de los subconjuntos de instrucciones podían requerir hasta 2 ½ veces más memoria que programas equivalentes con conjuntos de instrucciones complejos.
 - 2) La tasa de falla en memoria caché era considerablemente mayor en el caso de programas escritos en el subconjunto más sencillo.
 - 3) El tráfico por conductos de comunicación en el caso de programas escritos en el subconjunto de instrucciones más sencillo era aproximadamente el doble que en el caso de programas escritos en el subconjunto más complejo.

Estos resultados y otros parecidos indican que las arquitecturas RISC pueden tener consecuencias negativas en cuanto al desempeño.

9.12. Resumen de desempeño, Coprocesadores, RISC y flujo de datos (data flow)

En este módulo se analizan muchas de las técnicas de uso más frecuente para supervisar y evaluar el desempeño de los sistemas de cómputo. Tres objetivos básicos de la evaluación del rendimiento son la evaluación para selección, la proyección del rendimiento y la supervisión del desempeño.

- El *desempeño* es la eficiencia con la que el sistema cumple con sus objetivos. Algunas medidas comunes del desempeño son el tiempo de retorno, el tiempo de respuesta, la producción, la capacidad y la utilización.
- Los *tiempos* son útiles para realizar comparaciones rápidas entre equipos. Se ha usado a menudo el tiempo de suma de un computador o el número de sumas por segundo para comparar tiempos.
- Las *mezclas de instrucciones* emplean un promedio ponderado de diversos tiempos de instrucciones más adecuadas para una aplicación específica.
- Un *programa núcleo* es un programa representativo que podría ejecutarse en una instalación. Se cronometra para una máquina dada empleando las estimaciones de tiempos de instrucciones de fabricante, y así pueden hacerse comparaciones entre máquinas distintas de acuerdo con la velocidad de ejecución esperada del programa núcleo.
- Los *modelos analíticos* son representaciones matemáticas de sistemas de cómputo o de sus componentes. Existe un volumen considerable de resultados matemáticos que puede aplicar el evaluador para ayudar a estimar el desempeño de un sistema de cómputo dado.
- Un *banco de prueba* es un programa real que el evaluador ejecuta en el sistema de cómputo en evaluación. El evaluador conoce las características de rendimiento del banco de prueba en un equipo ya existente, de modo que cuando se ejecuta en un equipo nuevo el evaluador puede sacar conclusiones significativas.
- Los *programas sintéticos* son programas reales diseñados a la medida para ejercitar funciones específicas de un sistema de cómputo. Son útiles sobre todo cuando no existen bancos de prueba que realicen dichas funciones.
- La *simulación* es una técnica con la cual el evaluador desarrolla un modelo computarizado del sistema en evaluación. Luego el sistema se ejecuta en un sistema de cómputo, con lo que se refleja el comportamiento del sistema en evaluación. Los simuladores manejados por eventos son controlados por eventos que se producen en el simulador de acuerdo con distribuciones de probabilidades. Los simuladores manejados por libreto son controlados por datos derivados empíricamente y manipulados con cuidado de manera que reflejen el comportamiento esperado del sistema simulado.
- La *supervisión del desempeño* es la obtención y análisis de información acerca del rendimiento de sistemas ya existentes.
- Un recurso se convierte en *cuello de botella* y limita el desempeño total del sistema cuando no es capaz de manejar el trabajo que se le envía. Los recursos que operan cerca de su capacidad máxima tienden a *saturarse*; es decir, los procesos que compiten por la atención del recurso comienzan a interferir unos con otros. La hiperpaginación en un sistema paginado es un ejemplo clásico de saturación en sistemas operativos.
- En un *ciclo de retroalimentación*, las solicitudes que llegan disponen de información sobre el estado actual del sistema.
- En la *retroalimentación negativa*, la tasa de llegada de solicitudes nuevas puede disminuir como resultado de la información retroalimentada. La retroalimentación negativa contribuye a la estabilidad en sistemas manejados por colas.
- En sistemas con *retroalimentación positiva*, la información retroalimentada origina un aumento en algún parámetro. Esta retroalimentación puede provocar inestabilidad en sistemas manejados por colas. Los diseñadores deben incluir mecanismos de control de la retroalimentación positiva para evitar una operación inestable.
- Los *coprocesadores* son procesadores de aplicación especial que se agregan a los sistemas de cómputo para realizar operaciones que no están incluidas en el procesador o procesadores originales.

- Las *arquitecturas RISC* casi siempre tienen conjuntos de instrucciones en lenguaje de máquina poco abundantes, instrucciones relativamente sencillas, un gran número de registros, operaciones sencillas de carga y almacenamiento para transferir datos entre la memoria y los registros, conductos profundos y memorias caché. Aunque los programas RISC suelen ser más largos que sus “equivalentes” en CISC, casi siempre se ejecutan mucho más rápido.
- Aunque las *arquitecturas de flujo de datos* todavía no se han introducido en máquinas comerciales, las técnicas de flujo de datos se pueden emplear con provecho en compiladores que preparan a los programas objeto para ser ejecutados en otras arquitecturas en paralelo, como por ejemplo las máquinas con canalización.

9.13 Bibliografía recomendada para este Módulo

1. The Essence of measurement, Alan S. Morris. Prentice Hall, Englewood Cliff, NJ. 19956, 208 pages.
2. Object - oriented test & measurement software development in C++, Atchison / Lee / HP, Prentice Hall, Englewood Cliff, NJ. 1996, 416 pages.
3. The Benchmark Book, Richard Grace, Prentice Hall, Englewood Cliff, NJ. 1996, 500 pages.
4. Sistemas de Explotación de Computadores, CROCUS, Paraninfo 424 páginas 1987 (Capítulo 6)
5. Sistemas Operativos (Segunda Edición); H.M. Deitel., Addison Wesley - 938 páginas- 1993 - Capítulo 14; 1994. 827 páginas

GLOSARIO DE TÉRMINOS EN IDIOMA INGLÉS

Orgware
Kernel
Overhead
Enter

Time
Cycle
Benchmark

Task
Chips
Turnaround time

Instruction
Instruction mix
Response Time

American Standard Code for information interchange

GLOSARIO DE TÉRMINOS EN CASTELLANO

Medidas y modelos de sistema (Métrica)

Áreas de aplicación de las técnicas de evaluación

Modelos de Sistemas

Medidas sobre sistemas reales

Metodología de las medidas

Generalidades

Mecanismos cableados internos al sistema

Utilización de las medidas

Mejora de las prestaciones

Medidas de desempeño

Por qué es necesario supervisar y evaluar el desempeño

Mezcla de instrucciones

Modelos analíticos

Programas sintéticos

Supervisión del desempeño

Cuellos de botella y saturación

Retroalimentación negativa

Coprocesadores

Estudios cuantitativos

Métodos y medidas de evaluación

Objetivos de los modelos

Naturaleza de las medidas

Mecanismos de medida

Aparatos de medidas externas

Medidas programadas

Evaluación de los sistemas

Tendencias importantes que afectan los aspectos de desempeño

Técnicas para evaluar el desempeño

Tiempos

Programas núcleos (Kernels)

Juegos de ensayo y comparación (Benchmark)

Simulación

Whetstone, Dhrystone, LINPACK y Savage

Ciclos de retroalimentación

Retroalimentación positiva

Computación con un conjunto reducido de instrucciones (RISC)

ACRÓNIMOS USADOS EN ESTE MÓDULO

CISC	Complex Instruction Set Computer	RISC	Reduced Instruction Set Computer
CPU	Central Processing Unit	SW	SoftWare
S.O.	Sistema Operativo	HW	HardWare
MIPS	Millon of Instruction per second	E/S	Entrada / Salida
BIPS	Billon of Instruction per second (10^9)	TIPS	Trillon of Instruction per second (10^{12})
VLSI	Very Large Scale of Integration		

AUTOEVALUACIÓN DEL MODULO 9:

Preguntas:

- 1.- ¿Cuales son los objetivos de los Modelos de Sistemas?
- 2.- ¿Para que se necesita la evaluación y predicción del desempeño?
- 3.- ¿Que son las medidas de desempeño? ¿Que tipos conoce?
- 4.- Realice un cuadro de doble entrada con las ventajas y desventajas de las distintas técnicas para evaluar el desempeño
- 5.- ¿ Qué es un ciclo de retroalimentación y que tipos existen?
- 6.- ¿ Qué es el RISC y cuál es su ventaja y desventaja sobre el CISC?
- 7.- ¿Cómo se logra una buena medición cualitativa del sistema?
- 8.- Dentro de las técnicas de evaluación de desempeño ¿Qué malas consecuencias trae usar el tiempo como una técnica?
- 9.- ¿Por qué la simulación es tan costosa?
- 10.- ¿Qué son los benchmark?
- 11.- ¿Qué son los cuellos de botella?
- 12.- ¿Qué es productividad?
- 13.- ¿Qué son y cómo funcionan los coprocesadores?
- 14.- ¿ Qué son los ciclos de retroalimentación?
- 15.- Enumere las ventajas y desventajas de la evaluación de desempeño mediante el tiempo.

Múltiple Choice:

<p>1.- Los objetivos de los Modelos de Sistemas tratan de...</p> <ol style="list-style-type: none"> a) Medir las características de las peticiones y verificar la validez de las medidas b) Construir un modelo que contenga una cierta cantidad de parámetros ajustables para estudiar el comportamiento de un sistema cualquiera. c) Construir un modelo matemático o un simulador según el grado de exactitud que se necesite. d) Medir independientemente magnitudes relacionadas entre sí y verificar que se satisface dicha relación, e) Todas las anteriores f) Ninguna de las anteriores 	<p>2.- Sobre actividades de la CPU se puede medir....</p> <ol style="list-style-type: none"> a) IDLE o estados master, user b) Tiempo de ejecución c) Tiempo y frecuencia de interrupciones sobre CPU d) SYSTEM CALLS o cambios de modo del procesador (context switch) e) OVERHEAD Producido por el uso del procesador por el S.O. f) Frecuencia de programas g) Todas las anteriores h) Ninguna de las anteriores corresponden
<p>3.- Generalmente los mecanismos de medida para los datos elementales son:</p> <ol style="list-style-type: none"> a) Los contenidos, b) los sucesos, c) la compactación de los sucesos (frecuencias) 	<p>4.- Es necesario para efectuar cualquier medida...</p> <ol style="list-style-type: none"> a) Conocer el tiempo de ejecución b) Conocer el Tiempo y frecuencia de interrupciones sobre CPU c) Definir parámetros,

<p>d) Los volúmenes o cantidad de contenidos</p> <p>e) Todas las anteriores</p> <p>f) Ninguna de las anteriores</p>	<p>d) Definir precisión,</p> <p>e) Definir contexto.</p> <p>f) Definir las actividades de la CPU</p> <p>g) Todas las anteriores</p> <p>h) Ninguna de las anteriores corresponden</p>
<p>5.- Los métodos para validar las medidas son....</p> <p>a) multiplicar los puntos de medida de tal forma que se mida una misma magnitud de muchas formas independientes,</p> <p>b) medir independientemente magnitudes relacionadas entre sí y verificar que se satisface dicha relación,</p> <p>c) comparar los resultados con las precisiones de un simulador o de un modelo matemático,</p> <p>d) realizar medidas bajo una carga artificial de características muy simples, que permita calcular resultados por adelantado.</p> <p>e) Todas las anteriores</p> <p>f) Ninguna de las anteriores corresponden</p>	<p>6.- El problema de la evaluación de los sistemas se puede ver desde los puntos de vista...</p> <p>a) la velocidad de los trabajos</p> <p>b) De las tasas de utilización de los distintos recursos</p> <p>c) De la definición de las magnitudes que caracterizan a las prestaciones</p> <p>d) De la especificación de las características de la carga del sistema.</p> <p>e) Todas las anteriores</p> <p>f) Ninguna de las anteriores.</p>
<p>7.- Es necesario supervisar y evaluar el desempeño para....</p> <p>a) Reproducir las experiencias</p> <p>b) Supervisar del desempeño</p> <p>c) <i>Evaluación para selección de un nuevo equipamiento</i></p> <p>d) Proyectar el rendimiento del sistema</p> <p>e) Conocer el orden de magnitud de un problema</p> <p>f) Todas las anteriores</p> <p>g) Ninguna de las anteriores corresponden</p>	<p>8.- La medida del Desempeño (o rendimiento) es...</p> <p>a) La manera en la que un sistema de cómputo cumple con sus objetivos</p> <p>b) La eficiencia con la que lo hace.</p> <p>c) Una cantidad relativa más que absoluta</p> <p>d) Casi siempre estimada</p> <p>e) Utilizada como base para comparaciones.</p> <p>f) Todas las anteriores</p> <p>g) Ninguna de las anteriores.</p>
<p>9.- Los tipos de medidas del desempeño se caracterizan por...</p> <p>a) Fáciles de cuantificar: tales como la facilidad de uso.</p> <p>b) Difíciles de cuantificar: tales como el número de accesos a disco por minuto.</p> <p>c) Orientadas al sistema: como el tiempo de respuesta</p> <p>d) Orientadas al usuario: como el porcentaje de utilización del procesador.</p> <p>e) Todas las anteriores</p> <p>f) Ninguna de las anteriores.</p>	<p>10.- El funcionamiento global de un sistema se caracteriza por.....</p> <p>a) la velocidad de los trabajos</p> <p>b) las tasas de utilización de los distintos recursos</p> <p>c) La carga real del sistema que es reproducible.</p> <p>d) La carga artificial del sistema que es reproducible</p> <p>e) Todas las anteriores</p> <p>f) Ninguna de las anteriores.</p>
<p>11.- En la definición de las cargas artificiales se necesita que satisfagan las condiciones de....</p> <p>a) Ser representativas de las condiciones de explotación reales del sistema que se va evaluar, en el Sentido que se precise.</p> <p>b) Ser reproducibles.</p> <p>c) Ser fáciles de parametrizar.</p> <p>d) Todas las anteriores</p> <p>e) Ninguna de las anteriores.</p>	<p>12.- La técnica de mezclas de instrucciones emplea.....</p> <p>a) Un promedio ponderado de diversos tiempos de instrucciones más adecuado para una aplicación específica.</p> <p>b) Un promedio ponderado de tiempos para aquellas instrucciones de uso más frecuente en la instalación.</p> <p>c) Una simulación de la carga donde las instrucciones de varios tipos figuran con una determinada frecuencia</p> <p>d) Para poder caracterizar un tipo dado de aplicación.</p> <p>e) Todas las anteriores</p> <p>f) Ninguna de las anteriores.</p>
<p>13.- La Varianza de los tiempos de respuesta....</p> <p>a) Es una medida de la dispersión.</p> <p>b) Es una medida de su predecibilidad.</p> <p>c) Si es pequeña indica que los tiempos de respuesta que perciben los usuarios están en general cerca de la media.</p> <p>d) Si es considerable indica que algunos usuarios pueden estar recibiendo tiempos de respuesta muy distintos de la media.</p> <p>e) Todas las anteriores</p> <p>f) Ninguna de las anteriores.</p>	<p>14.- La Producción es la medida de...</p> <p>a) La cantidad de trabajo que se ha introducido en el sistema,</p> <p>b) La producción máxima que puede tener un sistema,</p> <p>c) La fracción de tiempo que está en uso algún recurso.</p> <p>d) El rendimiento de trabajo por unidad de tiempo.</p> <p>e) Todas las anteriores</p> <p>f) Ninguna de las anteriores.</p>
<p>15.- Entre los tipos de simuladores se tienen...</p> <p>a) Simuladores manejados por estadísticas.</p> <p>b) Simuladores manejados por los eventos.</p> <p>c) Simuladores manejados por libreto.</p> <p>d) Simuladores manejados por matemáticas.</p> <p>e) Simuladores manejados por neuronas.</p> <p>f) Todas las anteriores</p> <p>g) Ninguna de las anteriores.</p>	<p>16.- Los recursos que operan cerca de su capacidad máxima</p> <p>a) Se considera la fracción de tiempo que está en uso</p> <p>b) Se considera el rendimiento de trabajo por unidad de tiempo.</p> <p>c) Se consideran cuello de botella.</p> <p>d) Se considera porque tienden a saturarse.</p> <p>e) Todas las anteriores</p>

<p>17.- En situaciones de retroalimentación negativa....</p> <p>a) La tasa de llegada de las solicitudes nuevas puede disminuir como resultado de la retroalimentación de información.</p> <p>b) La tasa de llegada de las solicitudes nuevas puede aumentar como resultado de la retroalimentación de información.</p> <p>c) La tasa de llegada de las solicitudes nuevas no se altera como resultado de la retroalimentación de información.</p> <p>d) Contribuye a la estabilidad en los sistemas de colas.</p> <p>e) Contribuye a la inestabilidad en los sistemas de colas.</p> <p>f) Todas las anteriores</p> <p>g) Ninguna de las anteriores.</p>	<p>f) Ninguna de las anteriores.</p> <p>18.- La computación con un conjunto reducido de instrucciones (RISC)...</p> <p>a) Emplea un conjunto de instrucciones complejas de un procesador.</p> <p>b) Emplea un conjunto amplio de instrucciones individuales de lenguaje de máquina</p> <p>c) Emplea una pequeña porción de un conjunto de instrucciones de un procesador.</p> <p>d) Emplea un conjunto de instrucciones mejoradas que se utilizan con poca frecuencia</p> <p>e) Emplea un conjunto de instrucciones mejoradas que se utilizan con mucha frecuencia</p> <p>f) Todas las anteriores</p> <p>g) Ninguna de las anteriores.</p>
<p>19.- Las arquitecturas RISC pueden tener consecuencias negativas en cuanto al desempeño debido a....</p> <p>a) Los programas requieren más memoria que los programas equivalentes con conjuntos de instrucciones complejos.</p> <p>b) La tasa de falla en memoria caché es considerablemente mayor que en el caso de programas escritos con conjuntos de instrucciones complejos.</p> <p>c) El tráfico por conductos de comunicación es mayor que en el caso de programas escritos con el conjuntos de instrucciones complejos.</p> <p>d) Los programas escritos para máquinas RISC suelen tener más instrucciones por programa que los de las máquinas CISC.</p> <p>e) Todas las anteriores</p> <p>f) Ninguna de las anteriores.</p>	<p>20.- La utilización se caracteriza por</p> <p>a) Ser la fracción de tiempo que está en uso algún recurso.</p> <p>b) Que puede ser una medida engañosa.</p> <p>c) Que al parecer lo mejor es tener un porcentaje de utilización alto puede ser el resultado de un aprovechamiento ineficiente.</p> <p>d) Que una forma de lograr un alto porcentaje de utilización del procesador es ejecutar una serie de procesos, cada uno de los cuales está en un ciclo infinito.</p> <p>e) Todas las anteriores</p> <p>f) Ninguna de las anteriores.</p>

Respuestas a las preguntas:

1.- ¿Cuales son los objetivos de los Modelos de Sistemas?

Básicamente se busca cubrir los siguientes objetivos:

Construir un modelo que contenga una cierta cantidad de parámetros ajustables para estudiar el comportamiento de un sistema cualquiera.

Construir un modelo matemático o un simulador según el grado de exactitud que se necesite.

2¿Para que se necesita la evaluación y predicción del desempeño?

Se necesita durante:

- Los primeros momentos de a concepción y desarrollo de un sistema nuevo: el proveedor intenta predecir la naturaleza de las aplicaciones que se ejecutarán en el sistema, así como las cargas de trabajo que deben manejar dichas aplicaciones.
- En el desarrollo y realización del nuevo sistema sirven para determinar la mejor organización del equipo, las estrategias de administración de recursos que deben ponerse en práctica en el sistema operativo y si el sistema en evolución cumple o no con sus objetivos de rendimiento.
- En el lanzamiento al mercado el proveedor debe estar preparado para responder a preguntas de usuarios potenciales acerca de la capacidad del sistema para manejar ciertas aplicaciones con ciertos niveles de rendimiento.
- En la instalación tanto el usuario como el proveedor están interesados en obtener un desempeño óptimo. Se realizan ajustes finos para que el sistema funcione de la mejor manera posible dentro del ambiente de operación del usuario (proceso de afinación del sistema)

3.- ¿Que son las medidas de desempeño? ¿Que tipos conoce?

Las medidas absolutas de desempeño podría decirse que son cantidades tales como el número de trabajos por hora que puede atender un sistema de cómputo dado. Siempre que se estima una medida de desempeño lo normal es que sirva como base de comparaciones.

Los tipos de medidas que conozco son:

- Difíciles de cuantificar
- Fáciles de cuantificar
- Orientadas al usuario
- Orientadas al sistema

4.- Realice un cuadro de doble entrada con las ventajas y desventajas de las distintas técnicas para evaluar el desempeño

Técnica	Ventajas	Desventajas
Tiempos	-Con la llegada de familias de computadores se ofrecen computadoras con los cuales un usuario puede progresar a procesadores más rápidos conforme a sus necesidades. Los tiempos ofrecen una forma rápida de comparar entre sí los miembros de una familia de computadores. -Los tiempos se usan sobre todo para comparaciones rápidas	-Se requieren otras técnicas para evaluar en forma más significativa los complejos sistemas de hardware y software de la actualidad. En general sólo se hacen comparaciones de tiempos para unas cuantas operaciones básicas del equipo. Por tanto, al comparar procesadores para una aplicación específica, los tiempos ofrecen muy poca información.
Instruction Mix	-Las comparaciones entre máquinas pueden tener más validez que si se usan los tiempos por sí solos -Las mezclas son útiles en su mayor parte para comparaciones rápidas	-El evaluador debe advertir que conviene utilizar otras técnicas antes de tomar decisiones de adquisición vitales. Las ponderaciones empleadas en esta técnica tienden a ser muy subjetivas, por lo que es

		<p>preciso examinar con cuidado los puntos finos de los conjuntos de instrucciones de diferentes sistemas de cómputo</p> <ul style="list-style-type: none"> -Las mezclas ofrecen muy poca o ninguna información útil para evaluar programas -La complejidad del diseño de los equipos actuales hace cada vez más difícil establecer la validez de las mezclas de instrucciones. Con el empleo de memoria caché y canalización, una misma instrucción puede ejecutarse en tiempos diferentes, dependiendo de su contexto en las distintas ejecuciones.
Programas Núcleos	<ul style="list-style-type: none"> -Los núcleos producen mejores resultados que los tiempos o las mezclas de instrucciones -Una ventaja decisiva es que los núcleos son programas completos y, en última instancia, esto es lo que el usuario ejecuta en realidad es el sistema de cómputo considerado -Los núcleos pueden ser útiles para evaluar ciertos componentes de software de un sistema. 	<ul style="list-style-type: none"> -La preparación y cronometraje de los programas núcleos requieren un considerable esfuerzo manual
Modelos Analíticos	<ul style="list-style-type: none"> -Para los evaluadores con inclinaciones matemáticas, el modelo analítico puede resultar fácil de crear y modificar y ser una gran ayuda para estimar con rapidez y relativa precisión el desempeño de un sistema o componente de cómputo 	<ul style="list-style-type: none"> -Los evaluadores deben ser expertos y éstos no abundan en los ambientes de cómputo comercial -sólo existen soluciones claras para modelos más sencillos -Puede llevar a realizar suposiciones simplificadoras y destruir la utilidad y aplicabilidad del modelo
Juegos de Ensayo y Comparación	<ul style="list-style-type: none"> -El evaluador no tendrá que crear el banco de prueba, sólo necesita escogerlos de entre los programas de producción conocidos -No se realizan cronometrjes manuales de instrucciones individuales. El programa complejo se ejecuta con datos reales en la máquina real, de modo que el computador realiza la mayor parte del trabajo. -Se reducen al mínimo las posibilidades de error humano -El tiempo se puede medir con un cronómetro o lo puede medir el computador mismo. -Los bancos de prueba resultan, valiosos en ambientes con sistemas de multiprogramación, tiempo compartido, multiprocesamiento, bases de datos, comunicaciones de datos, y tiempo real porque se ejecutan en la máquina real en condiciones reales -Útiles para evaluar programas aun en ambientes de operación complejos -Son de utilidad para comparar el funcionamiento de un sistema antes y después de realizar modificaciones 	<ul style="list-style-type: none"> -Selección cuidadosa de una serie de bancos de prueba representativa de las características de trabajo en la instalación -No resultan útiles para predecir los efectos de cambios propuestos, a menos que exista otro sistema en el que se hayan incorporado las modificaciones y en el cual puedan ejecutarse los bancos de prueba
Programas Sintéticos	<ul style="list-style-type: none"> -Podrían no existir programas de banco de prueba para probar funciones de una nueva máquina 	<ul style="list-style-type: none"> -Los evaluadores no siempre tienen suficiente tiempo para codificar y depurar programas sintéticos

	-Útiles en ambiente de desarrollo	
Simuladores de Carga	<ul style="list-style-type: none"> -Es posible preparar un modelo de un sistema que no exista y después ejecutar el modelo para ver cómo se comportaría el sistema en ciertas circunstancias -Pueden evitar la construcción de sistemas mal diseñados poniendo en relieve los problemas en una etapa temprana del ciclo de desarrollo -Luego de establecer la validez de los modelos de simulación, su uso repetido puede resultar provechoso y económico 	<ul style="list-style-type: none"> -Requiere evaluadores con un alto nivel de conocimientos y puede consumir bastante tiempo de cómputo -Producen cantidades tremendas de datos que se deben analizar manualmente
Supervisión de Desempeño	<ul style="list-style-type: none"> -Resulta útil para averiguar lo bien que se comporta un sistema en términos de producción, tiempos de respuesta, predecible, etc. -Es posible localizar rápidamente los cuellos de botella y ayudar a los administradores a decidir cómo puede mejorarse el rendimiento -Los supervisores indican con precisión cómo está funcionando el sistema, y la gran cantidad de información producida puede ser en extrema valiosa, sobre todo en ambientes de desarrollo en los cuales quizá deban tomarse o modificarse decisiones clave de diseño con base en la operación observada del sistema -Los rastreos de ejecución de módulos muchas veces pueden poner al descubierto cuellos de botella. Un rastreo de ejecución de módulos muchas veces pueden poner al descubierto cuellos de botella. Un rastreo de ejecución de módulos podría mostrar que un pequeño subconjunto de los módulos se está usando gran parte del tiempo. Por lo tanto, si los diseñadores concentran sus esfuerzos de optimización en esos módulos, pueden mejorar en forma considerable el desempeño del sistema sin invertir trabajo y recursos en partes del sistema de uso poco frecuente 	<ul style="list-style-type: none"> -Los programas supervisores pueden distorsionar las lecturas de rendimiento porque ellos mismos consumen recursos del sistema -Los equipos supervisores son en general más costosos, pero con mínima influencia sobre la operación del sistema -Los supervisores producen enormes volúmenes de datos que deben analizarse, lo cual requiere considerables recursos de cómputo.

5.- ¿ Qué es un ciclo de retroalimentación y que tipos existen?

Es un caso en el que cierta información sobre el estado actual del sistema puede afectar a las solicitudes que lleguen posteriormente. Pueden ocurrir dos cosas, que las tasas de llegadas disminuyan o aumenten. Se define entonces como retroalimentación negativa al caso en que las tasas de llegadas disminuyen. Suele colaborar para evitar la congestión en un sistema de colas, distribuyendo equitativamente los procesos que van llegando. La retroalimentación positiva es el caso en el que las tasas de llegada pueden aumentar. Para implementarlo el sistema debe estar muy bien diseñado.

6.- ¿ Qué es el RISC y cuál es su ventaja y desventaja sobre el CISC?

El RISC es la computación con un conjunto reducido de instrucciones. Esto implica utilizar instrucciones lo mas simples posibles para facilitar el procesamiento de ellas. El CISC es lo contrario, es el uso de instrucciones complejas.

Ventajas de RISC sobre CISC:

- Las arquitecturas RISC suelen contar con un número elevado de registros y los compiladores RISC están diseñados para optimizar el código. Ambas cosas apuntan a ejecutar eficientemente.
- En RISC casi todas las instrucciones requieren de un ciclo de máquina a diferencia de en CISC.
- Se evita la microprogramación.
- Se utiliza la ramificación retardada. Después de una ramificación condicional ejecuta de todas maneras la instrucción, ganando así mucho tiempo.
- Los conjuntos de instrucciones RISC facilitan el trabajo del compilador.
- Tienden a usar varias memorias cache para agilizar la ejecución.
- Los programas RISC se ejecutan 2 o 3 veces más rápidos que los equivalentes CISC.
- Los diseños RISC se puede modificar fácilmente.
- Las instrucciones RISC ocupan palabras de memoria con longitud fija, lo que agiliza la ejecución de las mismas.
- Las instrucciones RISC casi siempre se ejecutan en 4 fases: obtención de la instrucción, lectura de registro, operación aritmético/lógica y escritura de registro.

Desventajas del RISC sobre el CISC:

- Los programas escritos para máquinas RISC suelen tener más instrucciones que sus semejantes aplicados sobre CISC.
- Las máquinas RISC utilizan mucho más los registros que los CISC pero lo malo es que los registros deben ser conservados y esto se refleja en una merma en la velocidad de ejecución debido a los cambios de contexto.
- Las operaciones en punto flotante se ejecutan más rápido en las CISC.

7.- ¿Cómo se logra una buena medición cualitativa del sistema?

Hay diferentes métodos de medir un sistema, de cualquiera de ellas, el mejor resultado del estudio les dará si combinan los métodos y aplicarlas en forma global (no en forma particular). Estos métodos se clasifican en dos grupos:

- Estudio de sistemas reales y programas
 - Medidas cableadas
 - Medidas programadas
- Representación de sistemas y programas
 - Modelos simulados
 - Modelos analíticos

8.- Dentro de las técnicas de evaluación de desempeño ¿Qué malas consecuencias trae usar el tiempo como una técnica?

Si bien estas hacen comparaciones por tiempo son rápidas, ya que se comparan mediante patrones de tiempos, no son confiables. Los patrones son tiempos promediados según operaciones básicas del equipo, pero si se compara mediante una aplicación específica los equipos comparados tendrán seguramente resultados diferentes, por lo que no da información confiable.

9.- ¿Por qué la simulación es tan costosa?

La simulación de carga es simplemente un programa, con la característica de que es parametrizada y utilizada en forma continua. En este se pueden estudiar los efectos del sistema, reflejando el comportamiento del sistema evaluado al modificar los parámetros.

Si bien hay varios tipos de simuladores (dos), no todos pueden aplicar una simulación para evaluar su sistema, por que requiere personal con un alto nivel de conocimiento y puede consumir mucho tiempo, al producirse grandes cantidades de datos estas deben ser analizadas. Todas estas características se resumen a una sola, costo (costo de tiempo, costo económico, etc.)

10.- ¿Qué son los benchmark?

Son conjuntos de programas considerados representativos para algún tipo de carga. Pueden ser programas reales u otros que fueron creados específicamente para esa prueba. Los resultados obtenidos producen comparaciones y dependiendo de eso se toma una decisión. Es una de las técnicas más usadas en las empresas para la toma de decisión para la compra de equipos.

11.- ¿Qué son los cuellos de botella?

Es cuando un recurso se encuentra totalmente ocupado realizando tareas y los demás recursos se encuentran libres o con capacidad de sobra. Un recurso que se convierte en cuello de botella es cuando está colmando su capacidad, se dice que el recurso está saturado. La forma de detectarlos es supervisando las colas de solicitudes de cada recurso, si empieza a crecer es porque están llegando más solicitudes que la capacidad del recurso.

12.- ¿Qué es productividad?

La definición de productividad en la empresa es: "Productividad es el cociente que se obtiene de dividir la producción por uno de los factores de la producción", aplicando esto a Métrica es la cantidad de tareas que se pueden desarrollar en una unidad de tiempo estipulada.

13.- ¿Qué son y cómo funcionan los coprocesadores?

Los coprocesadores son procesadores de aplicación especial, con funciones que quizá no tenga el procesador o que quizá tenga pero no realice bien.

Los coprocesadores más comunes que hay ahora son los coprocesadores de punto flotante, los cuales ejecutan operaciones comunes de punto flotante a muy altas velocidades en hardware; sin ellos dichas operaciones se ejecutan por lo regular en software (con relativa lentitud).

Los coprocesadores casi siempre funcionan como procesadores esclavos del procesador o procesadores centrales; el procesador principal asigna cálculos apropiados al coprocesador y recupera de inmediato el control cuando el coprocesador termina su trabajo.

Los coprocesadores de aplicación especial funcionan a veces como procesadores iguales con respecto a los procesadores principales; esto significa que los coprocesadores pueden trabajar en paralelo con los procesadores principales o unos con otros.

El ancho de banda entre la CPU y los coprocesadores pueden convertirse en un cuello de botella. Algunos coprocesadores de punto flotante se manejan como parte de la memoria para ayudar a aliviar el problema.

14.- ¿Qué son los ciclos de retroalimentación?

Un ciclo de retroalimentación es un caso en el que cierta información acerca del estado actual del sistema puede afectar a las solicitudes que llegan. Si la retroalimentación indica que dichas solicitudes pueden tener problemas para ser atendidas, quizá se les pueda enviar por otro conducto. La retroalimentación puede ser negativa, en cuyo caso pueden disminuir las tasas de llegada, o positiva, en cuyo caso pueden aumentar.

Retroalimentación Negativa

En situaciones de retroalimentación negativa, la tasa de llegada de las solicitudes nuevas puede disminuir como resultado de la retroalimentación de información.

En sistemas distribuidos, las salidas por spool a menudo pueden ser impresas por cualquiera de varios despachadores de impresoras equivalentes.

La retroalimentación negativa contribuye a la estabilidad de los sistemas de colas.

Retroalimentación Positiva

En situaciones de retroalimentación positiva, la tasa de llegada de nuevas solicitudes puede aumentar a causa de la retroalimentación de la información.

Los diseñadores de sistemas operativos deben tener cuidado al diseñar mecanismos con retroalimentación positiva para evitar que se presente una operación inestable. Es preciso supervisar los

efectos de cada modificación para ver si produce la mejora esperada. Si una modificación hace que se deteriore el rendimiento, esto indicará al sistema operativo que puede estar entrando en un intervalo inestable, de modo que se hace necesario ajustar las estrategias de asignación de recursos para continuar con una operación estable.

15.- Enumere las ventajas y desventajas de la evaluación de desempeño mediante el tiempo.

Las ventajas de evaluar un sistema a través del tiempo es que se puede lograr una rápida comparación entre miembros de una familia de computadoras. La obtención de resultados es rápida.

Las desventajas se centran en que este parámetro de comparación solo es útil para unas cuantas operaciones muy básicas, y no ofrece estimaciones reales en casos más complejos. La evaluación mediante tiempo requiere ser complementada con otras técnicas.

<i>Respuestas del múltiple choice.</i>

- | | | | | |
|------------|--------------|------------|--------------|------------|
| 1.- b, c. | 2.- g. | 3.- e. | 4.- c, d, e. | 5.- e. |
| 6.- c, d. | 7.- b, c, d. | 8.- f. | 9.- f | 10.- a, b. |
| 11.- d. | 12.- e. | 13.- e. | 14.- d. | 15.- b, c. |
| 16.- c, d. | 17.- a, d. | 18.- c, e. | 19.- d. | 20.- e |